POINT DENSITY-INVARIANT 3D OBJECT DETECTION AND POSE ESTIMATION

Su-A Kim[1, 2], **Kuk-Jin Yoon**[2]

[1]Intel Visual Computing Institute, Germany [2]Gwangju Institute of Science and Technology, Korea

https://sites.google.com/site/suakimpf/icip17

Introduction

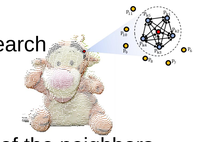
- It is crucial to extract distinctive and representative features of the objects and describe them efficiently.
- However, **point density variation** produces false correspondences and causes adverse effects on 3D object detection and pose estimation.

Goal:

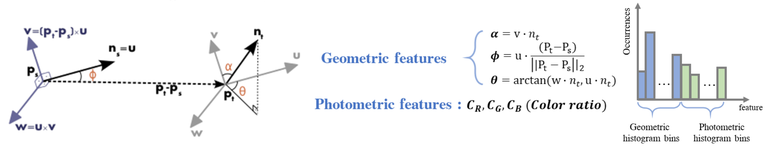
- Finding the correct correspondences under point density variations

Point Feature Histogram RGB (PFHRGB)[1]

- Best 3D feature descriptor
  - Have been evaluated as showing the best performance for 3D object and category recognition among 3D features [2].
- Neighbor definition
  - Define the neighbors around keypoints using a radius search



- Feature histogram description
  - Consider the relationship between all bidirectional pairs of the neighbors
  - Consist of angular and photometric features



- Normalization of the histogram increments to deal with point density variation → **BUT, NOT ENOUGH!**
  - By the number of bi-directional combinations of the neighbors $2 \cdot \binom{k}{2}$

References

[1] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments", KI-Künstliche Intelligenz, 2010

[2] L. A. Alexandre, "3D Descriptors for Object and Category Recognition: A Comparative Evaluation", IROS Workshops, 2012

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A01005455).

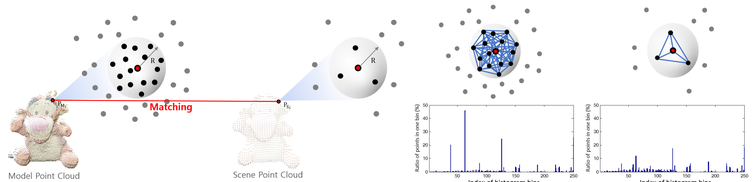
Since April 2017 the first author has been supported by the European Commission through the H2020-MSCA Distributed 3D Object Design (Grant No. 642841).

Point Density Variation

- Point density
  - The number of the points inside a fixed sphere when selecting a set of k-neighbor points around the keypoints
- Point density variation caused by two different situations

From different distances from the sensor — From different viewpoint

- Two points which are at the same position but having different density



Point density variation — At dense point cloud — At sparse point cloud

Proposed Method

- IDEA
  - Make the neighbors similar between two point clouds having the different density → Downsampling!!
  - We don't have any prior information about the point density of sparse point cloud.
    We can't guess appropriate scale for down-sampling the dense point cloud.
    → Multi-scale Feature Representation!
- **Multi-scale Feature Representation**

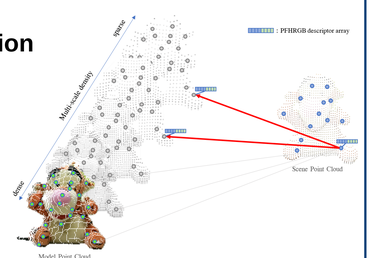
Keypoint detection on the model point cloud → Downsampling at coarse levels of scale → PFHRGB description at different scale levels

- **Scale Selection**

Similarity comparison of density on each keypoint → Scale selection having two smallest losses → Similarity comparison of PFHRGB descriptor

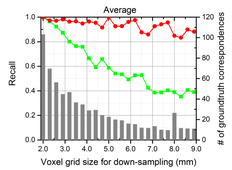
- **3D Object Detection and Pose Estimation**

Feature correspondence matching → Rejection of false correspondences → Initial pose estimation & Pose alignment → Object detection based on the refined pose

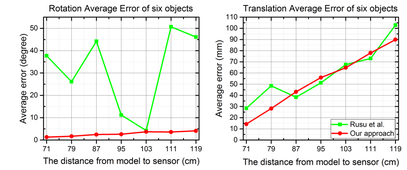


Experiments

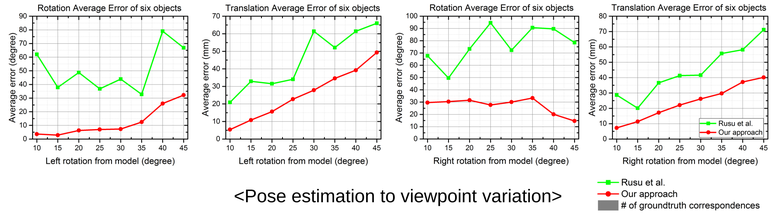
- Dataset
  - **Synthetic dataset** for the experiment of **feature matching** under point density variation
  - **Real dataset** for the experiment of **pose estimation** under the point density variation
  - **15 test scenes** including the point density variation, clutter and occlusion for **3D object detection**
- Experimental Results
  - Our approach outperforms Rusu *et al*. [1] showing the results of accurate feature matching, pose estimation, detection even if the point density variation exists.
  - The average computation time to estimate the correspondences and the initial pose of each object: Rusu *et al*. [1]: 2.161 sec, Our approach: 2.164 sec



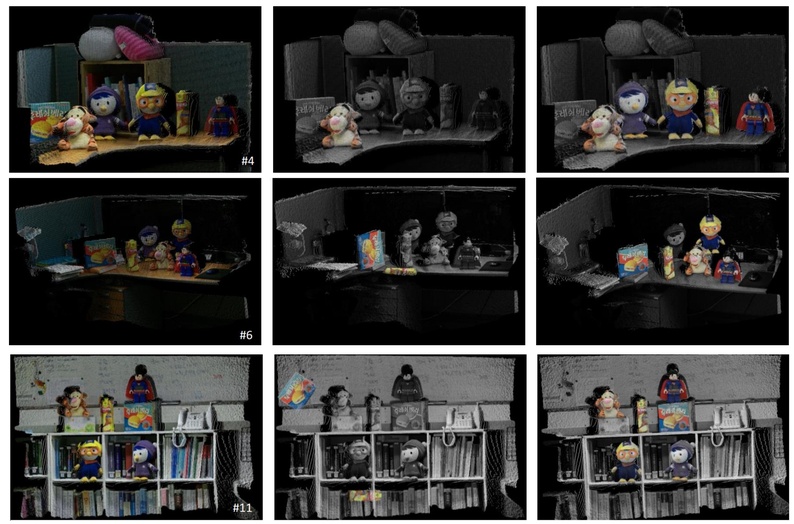
<Feature matching>



<Pose estimation to different distances from the sensor>



<Pose estimation to viewpoint variation>



Test scenes — Rusu *et al*. [1] — Proposed method

<3D object detection in general scenes>